IJECBE (2025), **3**, **4**, 693–711
Received (11 June 2025) / Revised (7 February 2026)
Accepted (9 February 2026) / Published (30 December 2025)
https://doi.org/10.62146/ijecbe.v3i4.145
https://ijecbe.ui.ac.id
ISSN 3026-5258

International Journal of Electrical, Computer and Biomedical Engineering

RESEARCH ARTICLE

Power Transformer Fault Identification Based on Random Forest and Data Resampling Considering Data Uncertainty

Elijah Chol Yom Majok* and Budi Sudiarto

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia
*Corresponding author. Email: elijah.chol@ui.ac.id

Abstract

Dissolved gas analysis (DGA) is a very important and reliable technique for fault identification in transformers to prevent grid outages. However, DGA datasets are sometimes imbalanced and also affected by uncertainty due to the presence of varying levels of noise, and this reduces the prediction accuracy of classification models. In this paper, we proposed a hybrid approach that combines the Random Forest classifier with multiple resampling techniques such as Random Over-Sampling, SMOTE, ADASYN, Borderline-SMOTE (versions 1 and 2), SMOTE-ENN, and SMOTE-Tomek. These methods were evaluated to identify the best combinations under different levels of uncertainty. Experiments were done on a publicly accessible DGA dataset with the addition of Gaussian noise (0%–20%) that simulates practical uncertainty caused by data measurement errors. The results indicate that SMOTE obtained the highest average accuracy of 82.46% and 81.29% with training-testing splits of 70:30 and 80:20, respectively. In addition to SMOTE, random oversampling achieved the highest average accuracy of 83.19% using a 90:10 split. These accuracy values are averages across all noise levels tested. The results suggest that appropriate selection of a resampling method improves fault identification of a random forest classifier under Gaussian noise.

Keywords: power transformers, random forest, dissolved gases, resampling, data imbalance

1. Introduction

Power transformers are critical equipment for the reliability of the power grid. When this equipment fails is due to malfunction, total blackout follows, and repair or replacement is very costly. Because of this, regularly maintaining the equipment is

highly recommended to prevent the occurrence of this problem. Dissolved gas analysis (DGA) is one of many tools used by power utilities to examine the internal condition of a transformer. A sample of insulating oil is taken, and the dissolved gases are extracted and analyzed to examine the internal condition of the transformer [1]. DGA is an advantageous diagnostic technique because it can be performed without taking the transformer out of service.

A new or normally operating transformer contains a small amount of hydrogen (H_2), methane (CH_4), ethane (C_2H_6), ethene (C_2H_4), and acetylene (C_2H_2) gases in insulating oil. But when there is a fault due to electrical or thermal stress, there is a rise in the concentrations of these gases. The presence and levels of these gases are indications of specific fault types, including corona, arcing, and overheating [2], [3]. To predict the potential presence of any of these faults, conventional methods such as the Key gas method, Doernenburg ratios, Roger's ratios, IEC codes, and the Duval triangle and pentagon are applied to interpret gas concentrations [4], [5], [6]. Although these methods are simple, they have low accuracy due to high data complexity and may easily lead to false diagnoses.

Artificial intelligence (AI) and machine learning (ML) techniques have recently been applied to overcome the limitations of conventional diagnostic methods. For example, [7] used a hybrid approach by combining fuzzy logic and particle swarm optimization to improve the diagnostic accuracy of Roger's four-ratio method and IEC 60599 codes. In [8], artificial neural networks (ANN) were implemented to classify incipient faults using DGA training data. The research article [9] introduced the PSO-ANN method that solves the limitations of the traditional ANN by optimizing ANN learning with the particle swarm optimization algorithm. A deep machine learning algorithm in [10] was proposed to learn the complex patterns in DGA data that limits the performance of conventional ANNs. Other machine learning techniques that have been applied to improve the diagnostic accuracy of the traditional DGA methods are decision trees [11], random forests [12], [13], k-nearest neighbors (KNN) [14], and support vector machines (SVM) [15]. Although these approaches have shown better fault diagnosis, they still struggle with the problems associated with data uncertainty due to measurement errors and imbalanced DGA datasets.

There are research papers that have proposed machine learning models that deal with data uncertainty and data imbalance. In [16], the paper proposed a hybrid gray wolf optimizer (HGWO) to refine fuzzy membership rules that can accommodate uncertainties in DGA data. A convolutional neural network (CNN) was proposed to improve diagnostic accuracy in the presence of noise in DGA data [17]. Many articles, on the other hand, have proposed data balancing methods to improve machine learning DGA-based fault classification [18], [19], [20], [21], [22], and [23].

Although there is extensive research carried out on data uncertainty and imbalance in DGAbased fault identification, the existing studies have solved these problems separately. Another limitation is that how different data balancing methods perform under varying degrees of data uncertainty has not yet been investigated. Therefore, the objective of this paper is to evaluate the performance of the random forest classification model under both data uncertainty and imbalance as follows:

1. Training and testing the random forest model on datasets with varying degrees of

uncertainty (0% to 20%).
2. Comparing the performance of different combinations of random forest and data balancing methods across a range of data uncertainty to find the best combination.

We organize the rest of this paper as follows: In Section 2, we briefly discuss dissolved gas analysis (DGA), data uncertainty, and data imbalance. In Section 3, we briefly discuss the methods used in the paper. These include resampling methods, the classification model, and dataset collection. Section 4 discusses the results and comparison of the proposed models with the previously used approaches in the article that used the same dataset. In Section 5, we draw the conclusions from the findings and give recommendations for possible future research directions.

2. Related Work

In this chapter, we discuss the key concepts and established theories that are used in this paper. Here, we present background information in a concise form so that understanding the materials related to the research and proper interpretation of the results will be clear and easy. The topics that are going to be discussed under this chapter are dissolved gas analysis (DGA), data uncertainty, and data imbalance.

2.1 Dissolved Gas Analysis (DGA)

Dissolved Gas Analysis (DGA) was first proposed between the late 1960s and the early 1970s as a diagnostic method for condition monitoring of oil-immersed power transformers. An oil sample is taken from transformer insulating oil, and the oil is tested for the presence of gases whose formation is an indication of a fault [24]. Thermal or electrical faults cause decomposition of the insulating oil which produces the formation of hydrocarbon gases such as hydrogen (H_2), methane (CH_4), ethane (C_2H_6), ethylene (C_2H_4), and acetylene (C_2H_2). Other gases that can also be detected during the testing include carbon monoxide (CO) and carbon dioxide (CO), which are the result of paper insulation decomposition. Oxygen (O_2) and nitrogen (N_2) may also be detected, but these gases are not caused by faults; they enter the oil from the atmosphere. The interpretation of gas composition can be helpful for identifying faults: low-temperature thermal faults (T1) produce CH_4 and C_2H_6 at temperatures below 300°C, medium-temperature faults (T2) form C_2H_4 at the temperature range of 300–700°C, and high-temperature faults (T3) produce C_2H_4 and traces of C_2H_6 above 700°C. Electrical faults such as partial discharges, lowenergy (D1), and high-energy discharges (D2) produce characteristic gas profiles including H_2 and CH_4 for corona (PD) and C_2H_2 and H_2 for arcing (D1 and D2). Combined (electrical and thermal) faults may produce a mix of all major fault gases.

2.2 Data Uncertainty

In any experimental process, ideal measurements are unattainable due to the presence of errors and uncertainties. When a measurement is done, there is some degree of uncertainty in the collected data [25]. Uncertainty refers to the range within which a true value may lie. Incorporating the concept of uncertainty is crucial in scientific research. It ensures that conclusions drawn from data are correct and reliable. When

uncertainty is ignored, the analysis done from that data will lead to misleading results and conclusions.

Uncertainty in measurements is caused by factors such as flawed measurement instruments, external conditions, and human error. Any measurement should try to minimize uncertainty as much as possible. In machine learning, understanding uncertainty is important because the accuracy of a machine learning model depends on the quality of the data used for training.

Gaussian noise was chosen because it reasonably approximates modeling uncertainty in experimental data. This is because, according to the Central Limit Theorem, when the sum of a large number of independent random variables with finite variance is taken, the sum tends toward a Gaussian distribution, and this does not depend on individual distributions of the contributing factors [26]. The noisy value of data can be determined in the equation (1).

$$X' = X + N(0, \sigma) \tag{1}$$

Where X' is a noisy value, X represents an original (true) value, $N(0, \sigma)$ shows normal distribution with mean 0, and σ represents a relative standard error from the actual value, and it can be calculated using the equation (2).

$$\sigma = X \times \frac{u}{100} \tag{2}$$

Where u is uncertainty percentage (randomly selected being either negative or positive, for example, u = ±5%)

2.3 Data Imbalance

Data imbalance is the uneven distribution of classes in a dataset. Data imbalance occurs when one or more classes have fewer samples than others. It is very common in classification tasks where faulty conditions are rare, such as fraud detection, medical diagnosis, fault detection in industrial systems, and many more. Machine learning models trained on highly imbalanced data tend to be biased toward the majority class [27]. This condition leads to low performance in classifying the minority class correctly. Imbalance severity of data can be quantified using the imbalance ratio (IR) in the equation (3).

$$IR_j = \frac{C_{max}}{C_j} \tag{3}$$

Where C_{max} represents the size of the highest class, and C_j is the size of the j^{th} class

3. Methodology

In chapter 3, we outline the research methodology used in the research. We briefly discuss the data collection and preparation, data resampling methods, and the random forest classification model.

3.1 Data Preparation

This research uses publicly accessible DGA datasets previously published in reputable peerreviewed journals [28]. The collected data will then be used to develop a random forest classifier for fault identification in power transformers. The total combustible gases (H_2, CH_4, C_2H_6, C_2H_4, C_2H_2) are used as input features for a random forest classifier to predict fault types (PD, D1, D2, T1, T2, T3). The faults are labeled as 1 to 6 since the machine learning model may not work properly with categorical outputs. This gas concentration will be used as input variables or features for the machine learning model, which predicts the corresponding fault type. The table below provides a summary of the collected DGA data.

Table 1. DGA dataset

Fault Identified	Number of Samples	Percentage (%)
PD	74	13
D1	91	15
D2	149	25
T1	111	19
T2	60	10
T3	104	18
Total	589	100

However, instead of using the raw gas concentrations directly, each gas concentration is converted into a percentage value relative to the total concentration of the combustible gases to normalize all the features. The conversion is shown in the equation (4).

$$P_i = \frac{G_i}{\sum_{i=1}^{5} G_j} \times 100\% \tag{4}$$

Where $\sum_{i=1}^{5}$ is the total concentration of all combustible gases (H_2, CH_4, C_2H_6, C_2H_4, C_2H_2), P_i represents the percentage of each gas.

3.2 Data Resampling Methods

Imbalanced datasets are a common challenge in classification tasks, particularly when the number of samples in one class significantly outweighs the other. Resampling is a widely adopted approach to solve this problem by modifying the distribution of data to achieve class balance. According to [27], a balanced dataset can improve the performance of classification models by reducing bias toward the majority class. Resampling techniques can be broadly categorized into three types: undersampling, oversampling, and hybrid methods.

3.2.1 Random Oversampling

Random oversampling increases minority class samples by randomly generating new samples from existing ones until the dataset is balanced. A balanced dataset can improve the ability of a model to learn minority class instances, and this may avoid bias during

training. Although this method is simple and able to balance datasets, it can result in overfitting [27]. By simply replicating existing samples, the model may become very reliable on these exact samples, and this can limit its ability to generalize to new data.

3.2.2 Synthetic Minority Over-Sampling Technique (SMOTE)

The Synthetic Minority Over-Sampling Technique (SMOTE) was introduced by Nitesh V. Chawla, and it is the most commonly used resampling method for balancing an imbalanced dataset [29]. SMOTE generates synthetic data points by interpolating between existing minority class data points. SMOTE first finds the k nearest neighbors for each minority class sample and then generates new synthetic samples by interpolating between the original sample and these neighbors.

3.2.3 Adaptive Synthetic Sampling (ADASYN)

Adaptive Synthetic Sampling (ADASYN) was developed by Haibo He et al. in 2008 [30]. The method was proposed to handle classification tasks where conventional resampling methods struggle. ADASYN generates synthetic minority class samples based on samples that are difficult to learn, unlike other oversampling methods. The method works on the principle that some minority class samples are difficult to classify because they cannot be easily distinguished from majority class samples.

3.2.4 Borderline-SMOTE

Borderline-SMOTE improves the original SMOTE by creating synthetic data from borderline samples that belong to the minority class [31]. The principle behind borderlineSMOTE is that misclassifications mostly occur near the decision boundary, making minority class samples close to the majority class that can be easily misclassified. The two variants of borderline-SMOTE are borderline-SMOTE1 and borderline-SMOTE2. Borderline-SMOTE1 generates synthetic samples by interpolating between a DANGER minority instance and any of its nearest neighbors within the minority class. Borderline-SMOTE2 creates synthetic samples by interpolating between a DANGER minority instance and only those of its nearest neighbors that are also classified as "danger" minority instances.

3.2.5 Hybrid Resampling Methods

The hybrid method technique combines oversampling and undersampling methods. SMOTE-Tomek and SMOTE-ENN are two of the most common resampling methods that use both oversampling and undersampling.

a) SMOTE-Tomek
   SMOTE-Tomek extends the SMOTE method by adding the Tomek Links method to data balancing. Tomek links identify pairs of nearest neighbors from different classes. A pair of links is formed when minority and majority samples are so close [32]. The method then removes instances that form links from the majority class. The SMOTE-Tomek method increases class separation by keeping data samples from the minority class from overlapping with the majority class samples, and this can improve classification performance.

b) SMOTE-ENN
SMOTE-ENN uses Edited Nearest Neighbors (ENN) instead of Tomek's links to remove samples that can be easily misclassified. ENN removes misclassified instances from the majority class and minority classes based on their nearest neighbors so that the decision boundary is clearly defined [33]. SMOTE-ENN is an aggressive resampling method, and as a result, it sometimes reintroduces imbalances into a dataset. Excessive deletion of samples can lead to loss of important information and low classification performance.

3.3 Random Forest Classifier

The random forest method, which is the most popular ensemble model, was introduced by Breiman in 2001 [34]. Random forest is a combination of multiple decision trees through a process called bagging. Each tree is trained from a bootstrap sample of training data, with data points being chosen at random. During tree growth, a random subset of features is split at each node. The combination of bootstrap sampling and random feature selection reduces model overfitting by preserving the diversity of the individual trees. Figure 1 shows the random forest classification process.

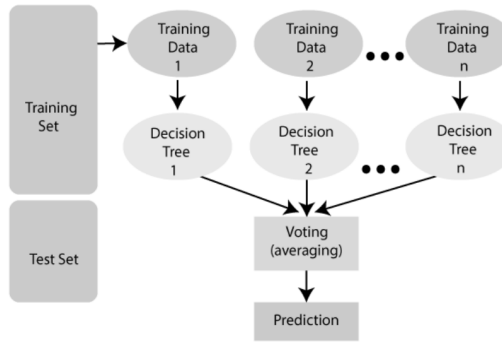

Figure 1. Random forest classifier

The theoretical characterization of random forest classifier can be mathematically represented. Let $\{h(x, \theta_k)\}_{k=1}^{K}$ be the ensemble of trees, where each θ_k is an i.i.d random vector encoding both bootstrapping and random-feature choices. The confidence level of the random forest model to correctly identify a class label is then calculated using margin function in equation (5).

$$mg(X, Y) = \frac{1}{K}\sum_{k=1}^{K} I(h(X, \theta_k) = Y) - \max_{j \neq Y} \frac{1}{K}\sum_{k=1}^{K} I(h(X, \theta_k) = j) \quad (5)$$

Where I shows an indicator function, with $I = \begin{cases} 1, \textit{if the condition is true} \\ 0, \textit{otherwise} \end{cases}$ X represents a feature vector, Y is a true class label for X, K is a total number of trees, $h(X, \theta_k)$ is a prediction of a single tree built using random parameter θ_k. After margin function is calculated, the next step involves finding the strength of a Random Forest which is

the average margin over the data distribution. Strength can be calculated using the equation (6).

$$s = \mathbb{E}_{X,Y}(mg(X, Y)) \tag{6}$$

Where $\mathbb{E}_{X,Y}$ denotes expected average margin over the joint distribution of input and labels In addition to strength, another factor called correlation is to be calculated. Correlation measures the average similarity in prediction errors between any two trees. Correlation is calculated using the equation (7).

$$\bar{\rho} = \frac{\mathbb{E}_{h_i h_j}(\rho(h_i(x), h_j(x))\sigma_{h_i}\sigma_{h_j})}{\mathbb{E}_{h_i,h_j}(\sigma_{h_i}\sigma_{h_j})} \tag{7}$$

Where $\mathbb{E}_{h_i h_j}$ denotes error between tree h_i and h_j,$\rho(h_i(x), h_j(x))$ is the correlation between the outputs of tree h_i and tree h_j , σ_{h_i} and σ_{h_j} are standard deviations associated with tree σ_{h_i} and tree σ_{h_j} . Strength and correlation give the following upper bound on the limiting generalization error in equation (8).

$$\epsilon = \frac{\bar{\rho}(1 - s^2)}{s^2} \tag{8}$$

Where ϵ denotes generalization error, $\bar{\rho}$ is a correlation, and s shows strength of trees. A smaller ρ (weakly correlated trees) and larger s (strong trees) both drive error downward.

The step-by-step process for random forest classification can be demonstrated below.

Algorithm Random Forest Classifier

Input:
- Training dataset $D = \{(x_1, y_1), ..., (x_n, y_n)\}$
- Number of trees K
- Number of features to sample at each split m ($m <$ total features M)

Output:
- Class $H\{h_{1(x)}, h_2(x), ..., h_K(x)\}$

Procedure:
- **for** $k = 1, 2, ..., K$ **do**
  - $D_k \leftarrow$BootstrapSample (D)
  - $T_k \leftarrow$GrowTree (D_k, m)
    - **while** stopping criteria not met **do**
      - Randomly select m features from M
      - Find best split among selected m features
      - Split node and repeat for child nodes
    - **end while**
  - $h_k(x) \leftarrow T_k$
- **end for**
- **Define** $H(x) \leftarrow$ majority vote
- **Return** $H(x)$

4. Results and Discussions

In this chapter, we present the results of the random forest classification model on imbalanced and balanced data under different data uncertainty conditions. The first step is to add uncertainty in the form of Gaussian noise to the dataset. The noise added is randomly generated, and it does change the class level, which may then influence model output. This means that there is no information transfer is introduced between training and testing samples, and this means that data leakage is extremely minimal. After that, the data is then split into training and testing subsets. The training samples are balanced using various resampling methods before they are used to train the model. The testing data is left untouched (not balanced) so that the data does not become artificial, which may produce false high performance. After the training dataset is balanced, both training and testing data are then converted to percentages. The random forest performance is then evaluated on both imbalanced and balanced datasets with various resampling methods to determine the best model based on classification accuracy, precision, recall, and F1 score.

4.1 Random Forest Classifier Performance Evaluation

This part presents an analysis of random forest classification accuracy in detail when combined with different resampling methods for each level of data uncertainty. In particular, the analysis provides a detailed account of how the classifier performs under different dataset training and testing percentage splits (split I: 70:30, split II: 80:20, and split III: 90:10). The simulation was run five (5) times, and the average was taken in order to account for variations in performances due to the randomness of uncertainty. All experiments were conducted using default hyperparameter settings across both imbalanced and balanced datasets, and all uncertainties. In other words, no additional hyperparameter tuning or optimization was performed. We did this deliberately to make sure that the influence of hyperparameters is controlled, and any observed performance variations are solely determined by the resampling methods and uncertainty levels.

4.1.1 Random Forest Classifier Performance at Split I

The performance of the Random Forest classifier in predicting fault classes under varying levels of data uncertainty is presented in Table 2 to Table 5. Specifically, Table 2 summarizes the classification accuracy and average accuracy across data uncertainty ranging from 0% to 20%, which is calculated for each resampling method. In addition to accuracy, other performance metrics, which include precision, recall, and F1 score, are presented in Table 3, Table 4, and Table 5, respectively, to test the overall strength of the model since it is dealing with data imbalance.

From the analysis, it can be observed that ROS produced the best performance results at low uncertainty levels (0% and 10%), which may be associated with the straightforward oversampling approaches being more suitable when data perturbations are minimal. At 5% uncertainty, the highest performance was obtained without applying any resampling technique, which implies that resampling may offer limited benefit under moderate uncertainty conditions.

Table 2. Classifier's performance representation using accuracy

Resampling Method	Accuracy at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	84.75	**83.62**	82.26	80.57	79.89	82.22
ROS	**85.31**	83.51	**83.28**	80.57	79.10	82.35
SMOTE	84.18	83.39	82.94	**81.02**	**80.79**	**82.46**
ADASYN	83.05	80.68	80.34	79.55	78.76	80.48
Borderline-SMOTE1	81.92	80.23	78.53	77.29	79.10	79.41
Borderline-SMOTE2	84.18	82.03	79.89	77.97	79.10	80.63
SMOTE-ENN	81.36	79.55	78.99	76.27	75.37	78.31
SMOTE-Tomek	84.18	83.17	82.94	80.79	80.57	82.33

Table 3. Classifier's performance representation using precision

Resampling Method	Precision at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	85.03	**84.08**	83.36	81.64	81.10	83.04
ROS	**85.59**	83.95	**84.28**	81.34	80.21	83.07
SMOTE	84.47	83.81	83.83	**81.57**	**81.69**	**83.07**
ADASYN	83.02	80.57	80.82	80.09	79.11	80.72
Borderline-SMOTE1	81.96	80.19	78.77	77.27	79.30	79.50
Borderline-SMOTE2	84.41	82.25	80.31	78.41	80.35	81.15
SMOTE-ENN	83.37	81.59	81.07	77.82	78.70	80.51
SMOTE-Tomek	84.30	83.84	83.68	81.74	81.58	83.03

Table 4. Classifier's performance representation using recall

Resampling Method	Recall at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	84.75	**83.62**	82.26	80.57	79.89	82.22
ROS	**85.31**	83.51	**83.28**	80.57	79.10	82.35
SMOTE	84.18	83.39	82.94	**81.02**	**80.79**	**82.46**
ADASYN	83.05	80.68	80.34	79.55	78.76	80.48
Borderline-SMOTE1	81.92	80.23	78.53	77.29	79.10	79.41
Borderline-SMOTE2	84.18	82.03	79.89	77.97	79.10	80.63
SMOTE-ENN	81.36	79.55	78.99	76.27	75.37	78.31
SMOTE-Tomek	84.18	83.17	82.94	80.79	80.57	82.33

When the uncertainty levels were increased to 15% and 20%, SMOTE performed better than the other methods, and this resulted in the highest average performance. This may be an indication that SMOTE is better at preserving data patterns under higher uncertainty than other resampling methods used. Across all uncertainty levels, SMOTE-ENN has shown the lowest results due to being more aggressive in data cleaning, and this makes it less robust to noisy datasets.

Table 5. Classifier's performance representation using F1 score

Resampling Method	F1 Score at Different Uncertanties (%)					
	0	5	10	15	20	Average
No Resampling	84.89	**83.85**	82.81	81.10	80.48	82.63
ROS	**85.45**	83.73	**83.78**	80.95	79.65	82.71
SMOTE	84.32	83.60	83.38	**81.29**	**81.24**	**82.77**
ADASYN	83.03	80.62	80.58	79.82	78.93	80.60
Borderline-SMOTE1	81.94	80.21	78.65	77.28	79.20	79.46
Borderline-SMOTE2	84.29	82.14	80.10	78.16	79.72	80.88
SMOTE-ENN	82.35	80.56	80.02	77.04	77.00	79.39
SMOTE-Tomek	84.24	83.50	83.31	81.26	81.07	82.68

4.1.2 Random Forest Classifier Performance at Split II

The classification results using accuracy, precision, recall, and F1 score metrics under different resampling methods and uncertainty levels are shown in Table 6 to Table 9. Under this data split, Borderline-SMOTE2 has the highest performance results when there is no data uncertainty applied (0%). However, its performance dropped significantly when uncertainty was applied to the dataset. This observation may be an indication that the method is more effective with the clean dataset but sensitive to noisy instances. When uncertainty was increased to 5%, the dataset balanced with ROS resampling provided the highest performance, which suggests that preserving the original data structure is more important under slightly higher uncertainty. At 10% uncertainty, the dataset with no resampling achieved the best results. However, its performance falls below that of SMOTE at 15% and 20% uncertainty. Like in the previous split (70:30), SMOTE-ENN consistently shows the lowest performance among the methods used, indicating reduced stability in the presence of noisy data. Under this data split (80:20), it can be observed that SMOTE has a better resistance to the influence of noise on classification performance.

Table 6. Classifier's performance representation using accuracy

Resampling Method	Accuracy at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	83.05	**83.56**	81.70	80.51	76.95	81.17
ROS	**83.05**	83.73	**80.34**	78.64	77.29	80.61
SMOTE	83.90	83.05	81.36	**80.68**	**77.46**	**81.29**
ADASYN	83.05	82.71	81.02	79.15	75.60	80.31
Borderline-SMOTE1	83.05	81.86	77.97	77.96	75.09	79.19
Borderline-SMOTE2	84.75	80.85	79.83	78.14	75.59	79.83
SMOTE-ENN	80.51	80.85	79.83	77.46	74.91	78.71
SMOTE-Tomek	81.36	83.39	80.85	79.83	76.10	80.31

Table 7. Classifier's performance representation using precision

Resampling Method	Precision at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	83.62	**84.70**	82.59	82.25	78.50	82.33
ROS	**83.62**	85.09	**81.51**	80.50	78.91	81.93
SMOTE	85.39	84.53	82.10	**82.43**	**79.31**	**82.75**
ADASYN	84.80	83.60	81.71	81.22	76.74	81.61
Borderline-SMOTE1	83.08	82.50	78.75	78.78	75.88	79.80
Borderline-SMOTE2	85.76	81.74	80.34	79.59	77.04	80.89
SMOTE-ENN	81.58	81.95	80.65	79.47	77.11	80.15
SMOTE-Tomek	82.42	84.60	81.63	81.20	77.74	81.52

Table 8. Classifier's performance representation using recall

Resampling Method	Recall at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	83.05	**83.56**	81.70	80.51	76.95	81.17
ROS	**83.05**	83.73	**80.34**	78.64	77.29	80.61
SMOTE	83.90	83.05	81.36	**80.68**	**77.46**	**81.29**
ADASYN	83.05	82.71	81.02	79.15	75.60	80.31
Borderline-SMOTE1	83.05	81.86	77.97	77.96	75.09	79.19
Borderline-SMOTE2	84.75	80.85	79.83	78.14	75.59	79.83
SMOTE-ENN	80.51	80.85	79.83	77.46	74.91	78.71
SMOTE-Tomek	81.36	83.39	80.85	79.83	76.10	80.31

Table 9. Classifier's performance representation using F1 score

Resampling Method	F1 Score at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	83.33	**84.16**	82.14	81.37	77.72	81.74
ROS	**83.33**	84.38	**80.92**	79.56	78.09	81.26
SMOTE	84.64	83.81	81.73	**81.55**	**78.40**	**82.03**
ADASYN	83.92	83.18	81.36	80.17	76.17	80.96
Borderline-SMOTE1	83.06	82.19	78.36	78.37	75.48	79.49
Borderline-SMOTE2	85.25	81.26	80.08	78.86	76.41	80.37
SMOTE-ENN	81.04	81.38	80.24	78.45	75.99	79.42
SMOTE-Tomek	81.89	84.00	81.24	80.51	76.91	80.71

4.1.3 Random Forest Classification Performance at Split III

The classification performance under this split (90:10) is presented in Table 10 to Table 13. With this data split, ROS achieves the best results in almost all data uncertainties except at 20where it scores lower than the dataset with no resampling technique applied. The other observation made is that ROS showed the same performance when there was a clean dataset, as the SMOTE-Tomek method for accuracy and recall; However, its precision is better than that of SMOTE-Tomek. This may be due to its ability to minimize data distortion and maintain data structure, which led to

better class discrimination. Unlike the previous splits, SMOTE-ENN performance increased, whereas Borderline-SMOTE1 significantly dropped to the lowest of all the methods. This may indicate that the aggressive elimination of instance overlaps usually associated with SMOTE-ENN is lower with the increased training dataset. Another important observation made is that ADASYN and SMOTE-Tomek showed similar performance in terms of accuracy and recall in split I, but SMOTE-Tomek produced better precision results. However, in the splits II and III, SMOTE-Tomek produced higher results than ADASYN. This is likely due to the increased presence of "hard-to-classify" instances generated by ADASYN, which distorted class distinction.

Table 10. Classifier's performance representation using accuracy

Resampling Method	Accuracy at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	86.44	**84.41**	82.71	80.34	78.65	82.51
ROS	**88.14**	86.10	**83.39**	82.03	76.27	83.19
SMOTE	86.44	83.05	81.02	**81.70**	**77.29**	**81.90**
ADASYN	84.75	81.02	77.97	78.31	74.24	79.26
Borderline-SMOTE1	79.66	79.66	77.63	77.96	74.24	77.83
Borderline-SMOTE2	79.66	80.00	81.36	78.99	73.22	78.64
SMOTE-ENN	84.75	81.70	78.65	80.00	73.22	79.66
SMOTE-Tomek	88.14	85.09	82.04	81.02	76.95	82.65

Table 11. Classifier's performance representation using precision

Resampling Method	Precision at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	87.19	**84.89**	83.31	81.07	79.78	83.25
ROS	**89.26**	86.74	**84.45**	82.92	77.30	84.13
SMOTE	86.89	83.33	81.02	**82.31**	**78.35**	**82.38**
ADASYN	85.11	81.85	79.55	79.08	75.23	80.16
Borderline-SMOTE1	79.73	79.86	77.92	78.78	75.86	78.43
Borderline-SMOTE2	81.71	80.24	81.55	79.89	75.10	79.70
SMOTE-ENN	85.09	82.55	79.92	81.57	75.47	80.92
SMOTE-Tomek	88.32	85.41	82.47	82.31	78.44	83.39

In addition to the use of tables to represent accuracy, the performance is presented as graphs and confusion matrices in Figure 2 to Figure 8. The confusion matrices summarize the classification accuracy of the random forest classifier. Each confusion matrix displays the performance for the best resampling method for each training and testing split. The average accuracy classification of the results is calculated for each fault in each confusion matrix across the various uncertainty levels. The results are then represented with one confusion matrix.

Figure 6 presents the confusion matrix average accuracy of SMOTE for each fault type under split I. The classifier obtained relatively high prediction accuracy for T1 (91.42%), T3 (88.77%), and D1 (88.63%) faults. Moderate accuracy is observed for

Table 12. Classifier's performance representation using recall

Resampling Method	Recall at Different Uncertainties (%)					
	0	5	10	15	20	Average
No Resampling	86.44	**84.41**	82.71	80.34	78.65	82.51
ROS	**88.14**	86.10	**83.39**	82.03	76.27	83.19
SMOTE	86.44	83.05	81.02	**81.70**	**77.29**	**81.90**
ADASYN	84.75	81.02	77.97	78.31	74.24	79.26
Borderline-SMOTE1	79.66	79.66	77.63	77.96	74.24	77.83
Borderline-SMOTE2	79.66	80.00	81.36	78.99	73.22	78.64
SMOTE-ENN	84.75	81.70	78.65	80.00	73.22	79.66
SMOTE-Tomek	88.14	85.09	82.04	81.02	76.95	82.65

Table 13. Classifier's performance representation using F1 score

Resampling Method	F1 Score at Different Uncertainties(%)					
	0	5	10	15	20	Average
No Resampling	86.81	**84.65**	83.00	80.70	79.21	82.87
ROS	**88.70**	86.42	**83.92**	82.47	76.78	83.66
SMOTE	86.66	83.19	81.02	**82.00**	**77.82**	**82.14**
ADASYN	84.93	81.43	78.75	78.69	74.73	79.61
Borderline-SMOTE1	79.69	79.75	77.77	78.37	75.04	78.12
Borderline-SMOTE2	80.67	80.12	81.45	79.44	74.15	79.17
SMOTE-ENN	84.91	82.12	79.28	80.78	74.33	80.28
SMOTE-Tomek	88.23	85.25	82.25	81.66	77.79	83.04

T2 (78.22%), PD (72.18%), and D2 (70.79%) faults. The D2 fault displays the highest misclassification, with 29.21% of instances misclassified as PD (12.48%), D1 (7.46%), and T1 (9.27%). The lowest prediction accuracy for the D2 fault suggests that the classifier struggles to correctly differentiate the D2 fault type from the other faults.

Figure 7 displays the classifier performance with the SMOTE confusion matrix for split II. The T2 fault achieved the highest accuracy, correctly identifying 88% of the samples. Moderate accuracy results were also observed for faults T1 (86.84%), D1 (84.26%), and T3 (83.24%). In addition, a significant reduction in accuracy occurred for the D2 fault, with only 75.06% correctly identified and 24.94% misclassified. The lowest classification was recorded by the PD fault, identifying 69.91% correctly and misclassifying 30.09%, with 24.54% incorrectly labeled as D1, 4% labeled as D2, and 1.55% as T1.

Figure 8 shows the random forest classifier confusion matrix with ROS at split III of the dataset. The figure shows the average performance for the different fault types. The model recorded relatively high accuracy for D1 (93.87%), T1 (88.55%), and T3 (87.60%) faults. Moderate accuracy is observed for PD, D2, and T2 faults, with correct predictions ranging from 72% to 78%. The most significant misclassifications occur for T2 (27.61%), PD (23.11%), and D2 (22%) faults. The high false classifications are caused by the difficulty of the classifier to differentiate between these fault types. This difficulty is likely caused by noisy instances which create increased class overlaps.

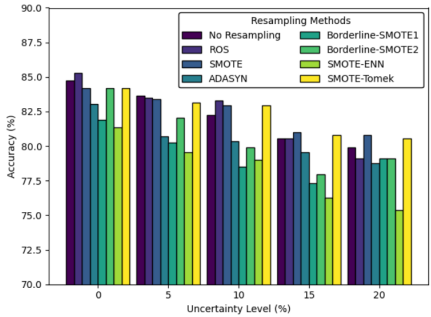

Figure 2. RF-SMOTE accuracy at split I

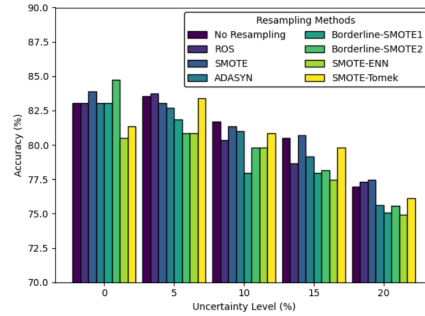

Figure 3. RF-SMOTE accuracy at split II

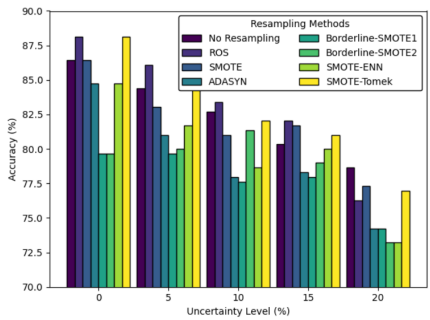

Figure 4. RF-ROS accuracy at split III

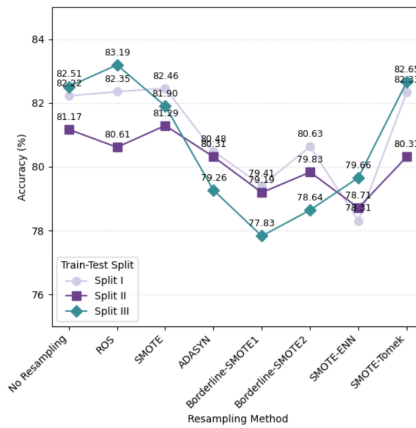

Figure 5. Average accuracy across each split

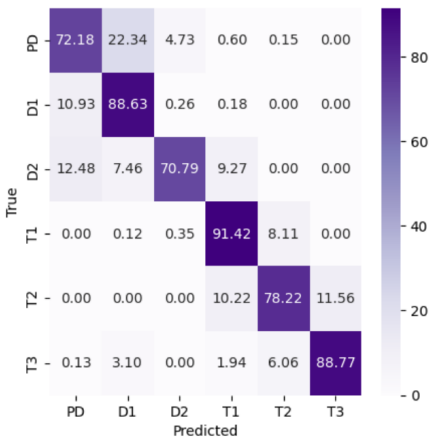

Figure 6. RF-SMOTE confusion matrix for split I

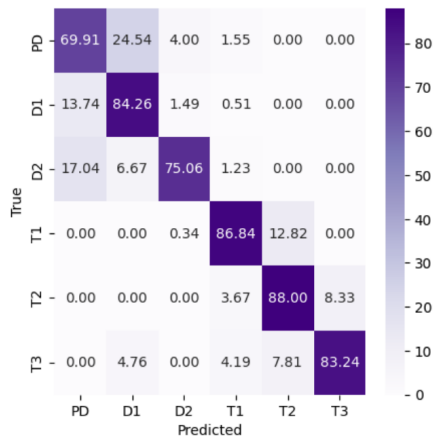

Figure 7. RF-SMOTE confusion matrix for split II

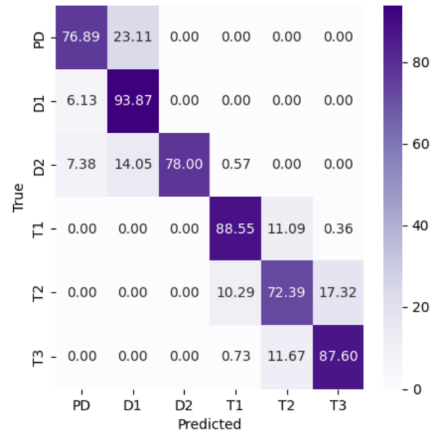

Figure 8. RF-ROS confusion matrix for split III

4.2 Comparison with a Previous Study

The DGA dataset used in this research has been used before in the literature. For example, P.A.R. Azmi et al. applied various resampling techniques with different machine learning classifiers [20]. Among all the methods used, the support vector machine (SVM) with edited nearest neighbors (ENN) achieved the highest classification accuracy. However, this study further improved the performance using a random forest classifier with the resampling methods. A detailed comparison of the classification accuracies between the proposed approach and that of P.A.R. Azmi et al. is presented in the table below.

Table 14. Comparison between the previous methods and the proposed methods

Train-Test (%)	Previous Method		Proposed Methods	
	Method	Accuracy (%)	Method	Accuracy (%)
70:30	SVM-ENN	77.33	RF-ROS	85.31
80:20	SVM-ENN	78.00	RF-Borderline-SMOTE2	84.75
90:10	SVM-ENN	88.00	RF-ROS	88.14

In the table above, the proposed approach provided increases in accuracy compared to the previous study as follows: 7.98% with RF-ROS for split I (70:30), 6.75% with RF-borderlineSMOTE2 for split II (80:30), and 0.14% with RF-ROS for split III (90:10). To clarify, the analysis only compares the results with the previous study for each training and testing split of the dataset without added uncertainty. This is because the previous study did not account for data uncertainty in the analysis.

5. Conclusion

In this paper, we presented the performance of random forest (RF) combined with different resampling methods to predict faults in power transformers when data imbalance and uncertainty are considered. Among all the models studied, RF combined with SMOTE obtained the highest average accuracy with training and testing splits of 70:30 and 80:20. In addition, RF with ROS achieved the highest average accuracy with a 90:10 split. These results suggest that the choice of a resampling method with a random forest classification model for imbalanced and uncertain data is crucial. This suggests that RF with suitable resampling methods provides satisfactory results and can be used in utilities for power transformer fault diagnosis.

Future work should investigate conventional methods such as Duval triangles and pentagons, IEC code ratios, and Rogers' ratios. In addition to that, future research should use the random forest classifier with other machine learning models, such as SVM, KNN, ANN, and so on, to obtain complete and valid results.

References

[1] *CIGRE 761 - Condition Assessment of Power Transformers*. Tech. rep. CIGRE, 2019.

[2] *IEEE Guide for the Interpretation of Gases Generated in Mineral Oil-Immersed Transformers*. IEEE C57.104-2019. 2019.

[3] *Mineral oil-filled electrical equipment in service - Guidance on the interpretation of dissolved and free gases analysis*. IEC 60599. 2022.

[4] R. R. Rogers. "IEEE and IEC Codes to Interpret Incipient Faults in Transformers". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 13.5 (1978), pp. 349–354. DOI: 10.1109/TEI.1978.298141.

[5] M. Duval. "A review of faults detectable by gas-in-oil analysis in transformers". In: *IEEE Electrical Insulation Magazine* 18.3 (2002), pp. 8–17. DOI: 10.1109/MEI.2002.1014963.

[6] M. L. L. Duval. "The Duval Pentagon - A new complementary tool for the interpretation of dissolved gas analysis in transformers". In: *IEEE Electrical Insulation Magazine* 30 (2014), p. 9. DOI: 10.1109/MEI.2014.6943428.

[7] I. B. M. Taha, A. Hoballah, and S. S. M. Ghoneim. "Optimal ratio limits of Rogers' four-ratios and IEC 60599 code methods using particle swarm optimization fuzzy-logic approach". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 27.1 (2020), pp. 222–230. DOI: 10.1109/TDEI.2019.008395.

[8] S. S. M. Ghoneim, I. B. M. Taha, and N. I. Elkalashy. "Integrated ANN-based proactive fault diagnostic scheme for power transformers using dissolved gas analysis". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 23.3 (2016), pp. 1838–1845. DOI: 10.1109/TDEI.2016.005301.

[9] F. et al. Guerbas. "Neural networks and particle swarm for transformer oil diagnosis by dissolved gas analysis". In: *Scientific Reports* 14.1 (2024). DOI: 10.1038/s41598-024-60071-0.

[10] L. Jin et al. "Deep Machine Learning-Based Asset Management Approach for Oil-Immersed Power Transformers Using Dissolved Gas Analysis". In: *IEEE Access* 12 (2024), pp. 27794–27809. DOI: 10.1109/ACCESS.2024.3366905.

[11] S. R. Al-Sakini et al. "Dissolved Gas Analysis for Fault Prediction in Power Transformers Using Machine Learning Techniques". In: *Applied Sciences* 15.1 (2025). DOI: 10.3390/app15010118.

[12] Suwarno et al. "Machine learning based multi-method interpretation to enhance dissolved gas analysis for power transformer fault diagnosis". In: *Heliyon* 10.4 (2024). DOI: 10.1016/j.heliyon.2024.e25975.

[13] R. A. et al. Prasojo. "Precise transformer fault diagnosis via random forest model enhanced by synthetic minority over-sampling technique". In: *Electric Power Systems Research* 220 (2023). DOI: 10.1016/j.epsr.2023.109361.

[14] O. Kherif et al. "Accuracy Improvement of Power Transformer Faults Diagnostic Using KNN Classifier With Decision Tree Principle". In: *IEEE Access* 9 (2021), pp. 81693–81701. DOI: 10.1109/ACCESS.2021.3086135.

[15] R. F. R. B. Souza. "Dissolved Gas Analysis to Identify Faults and Improve Reliability in Transformers Using Support Vector Machines". In: *2016 Clemson University Power Systems Conference (PSC)*. 2016. DOI: 10.1109/PSC.2016.7462827.

[16] A. Hoballah, D. E. A. Mansour, and I. B. M. Taha. "Hybrid Grey Wolf Optimizer for Transformer Fault Diagnosis Using Dissolved Gases Considering Uncertainty in Measurements". In: *IEEE Access* 8 (2020), pp. 139176–139187. DOI: 10.1109/ACCESS.2020.3012633.

[17] I. B. M. Taha, S. Ibrahim, and D. E. A. Mansour. "Power transformer fault diagnosis based on DGA using a convolutional neural network with noise in measurements". In: *IEEE Access* 9 (2021), pp. 111162–111170. DOI: 10.1109/ACCESS.2021.3102415.

[18] H. C. Chen, Y. Zhang, and M. Chen. "Transformer Dissolved Gas Analysis for Highly-Imbalanced Dataset Using Multiclass Sequential Ensembled ELM". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 30.5 (2023), pp. 2353–2361. DOI: 10.1109/TDEI.2023.3280436.

[19] K. N. V. P. S. Rajesh et al. "Influence of Data Balancing on Transformer DGA Fault Classification With Machine Learning Algorithms". In: *IEEE Transactions on Dielectrics and Electrical Insulation* 30.1 (2023), pp. 385–392. DOI: 10.1109/TDEI.2022.3230377.

[20] P. A. R. Azmi, M. Yusoff, and M. T. M. Sallehud-din. "Improving transformer failure classification on imbalanced DGA data using data-level techniques and machine learning". In: *Energy Reports* 13 (2025), pp. 264–277. DOI: 10.1016/j.egyr.2024.12.006.

[21] J. et al. Chen. "A novel method for power transformer fault diagnosis considering imbalanced data samples". In: *Frontiers in Energy Research* 12 (2024). DOI: 10.3389/fenrg.2024.1500548.

[22] A. Dhini et al. "Data-driven fault diagnosis of power transformers using dissolved gas analysis (DGA)". In: *International Journal of Technology* 11.2 (2020), pp. 388–399. DOI: 10.14716/ijtech.v11i2.3625.

[23] L. Wang, T. Littler, and X. Liu. "Hybrid AI model for power transformer assessment using imbalanced DGA datasets". In: *IET Renewable Power Generation* 17.8 (2023), pp. 1912–1922. DOI: 10.1049/rpg2.12733.

[24] B. Vahidi and A. Teymouri. *Quality Confirmation Tests for Power Transformer Insulation Systems*. Springer, 2019. DOI: 10.1007/978-3-030-19693-6.

[25] S. van der Leeuw. "Uncertainties". In: *Interdisciplinary Contributions to Archaeology*. Springer, 2016, pp. 157–169. DOI: 10.1007/978-3-319-27833-9_9.

[26] *Central Limit Theorem - an overview*. https://www.sciencedirect.com/topics/mathematics/central-limit-theorem. Accessed: 2026-01-30.

[27] R. Ghorbani and R. Ghousi. "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques". In: *IEEE Access* 8 (2020), pp. 67899–67911. DOI: 10.1109/ACCESS.2020.2986809.

[28] *DGA Dataset (589)*. https://github.com/Saleh860/DGA/blob/master/datasets/dataset_(589).xlsx. Accessed: 2025-06-06.

[29] N. V. Chawla, K. W. Bowyer, and L. O. Hall. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[30] H. He et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *IEEE International Joint Conference on Neural Networks*. 2008, pp. 1322–1328. DOI: 10.1109/IJCNN.2008.4633969.

[31] H. Han, W.-Y. Wang, and B.-H. Mao. "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning". In: *Advances in Intelligent Computing*. Springer, 2005, pp. 878–887.

[32] I. Tomek. "Two Modifications of CNN". In: *IEEE Transactions on Systems, Man, and Cybernetics* (1976), pp. 769–772.

[33] D. L. Wilson. "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data". In: *IEEE Transactions on Systems, Man, and Cybernetics* 2.3 (1972), pp. 408–421. DOI: 10.1109/TSMC.1972.4309137.

[34] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.